**Ascensor controlado por una FSM**



López de Meneses Calvo, Laura

Ortega Monge, Maria 54777

Villajos Mateo,Laura

**Descripción del trabajo**

Teniendo como base, la información que nos han impartido este curso en la asignatura de Sistemas Electrónicos Digitales, vamos a realizar la descripción hardware en vhdl del funcionamiento de un ascensor.

El proceso de diseño contara con todas las fases necesarias.

* Análisis de requisitos:

Para nuestro diseño queremos implementar un ascensor con un número determinado de pisos, en este caso 7, que cuando entre una persona espere a que se presione un botón, una vez pulsado ese botón será nuestro destino y a menos que entre otra persona( lo que activaría el sensor de presencia y se volverían a abrir las puertas), se cerraran las puertas y subirá o bajara los pisos determinados hasta llegar al destino, finalmente se abrirán las puertas para que las personas puedan bajarse y se quedará en reposo en ese piso.

Durante ese proceso de subida y bajada estará activa la salida motora en condiciones de subida y bajada.

Aquí va el diagram de estafos

* División funcional del diseño:

**Desarrollo a un nivel no trivial**

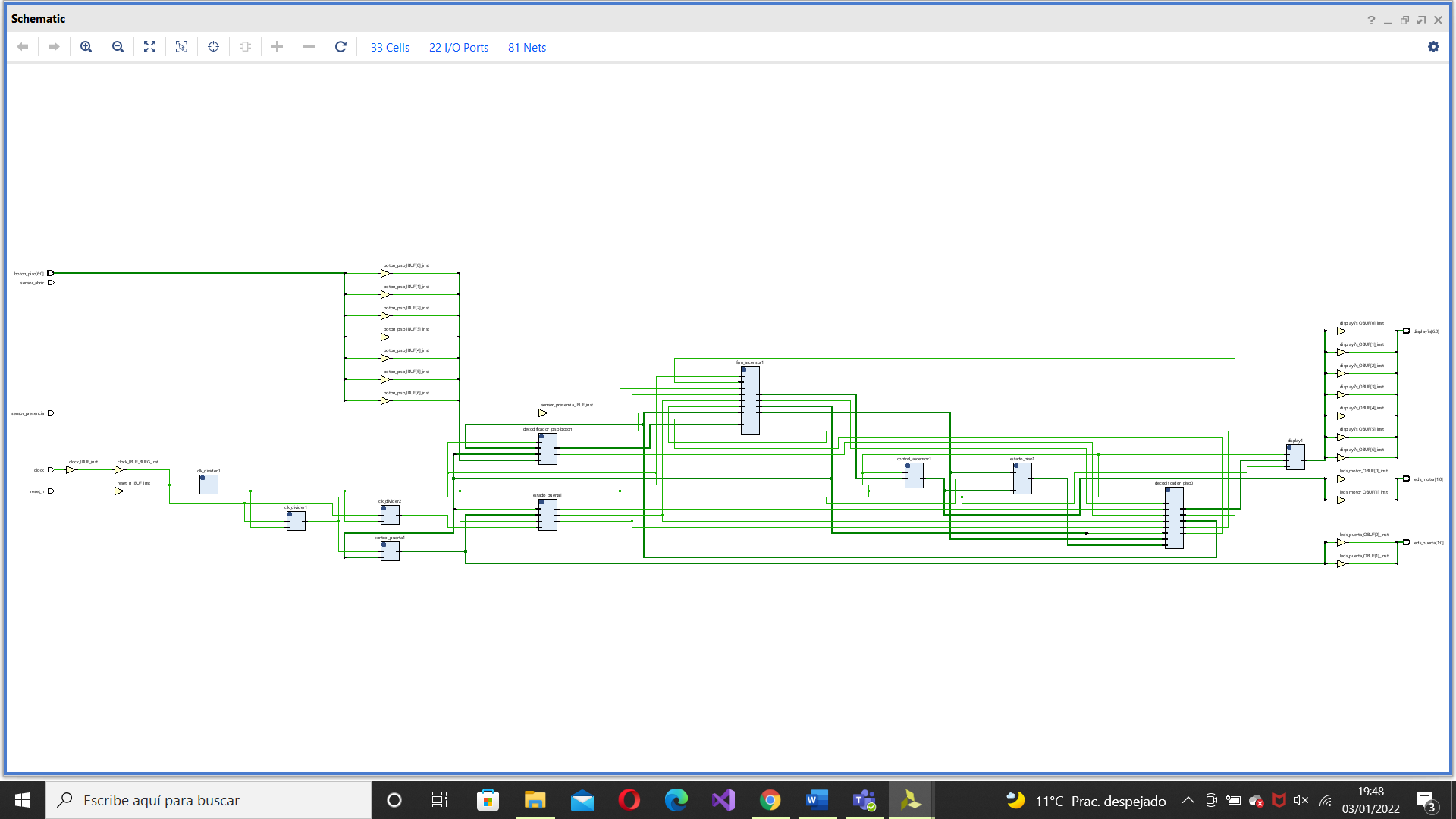
Es necesario dividir el diseño total como pequeñas funcionalidades más básicas, de tal forma que abordar la codificación de cada elemento se remonte a la funcionalidad más simple posible.

Aquí va cada bloque pequeñito y su descripcion

**Arquitectura del sistema**

En la arquitectura del sistema vemos como se conectan todas esas funcionalidades

antes descritas, pasamos de ver un proyecto formado por varios componentes a un componente global, en nuestro caso TOP\_ASCENSOR1.



Al ser tantos componentes e interconexiones hemos hecho una a mano para que se vea mejor.

Aquí va la top hecha a mano nuestra

* Verificación funcional

Una vez hemos realizado nuestro primer intento de diseño tenemos que comprobar que el diseño realmente cumple los requisitos establecidos y se amolda a lo deseado.

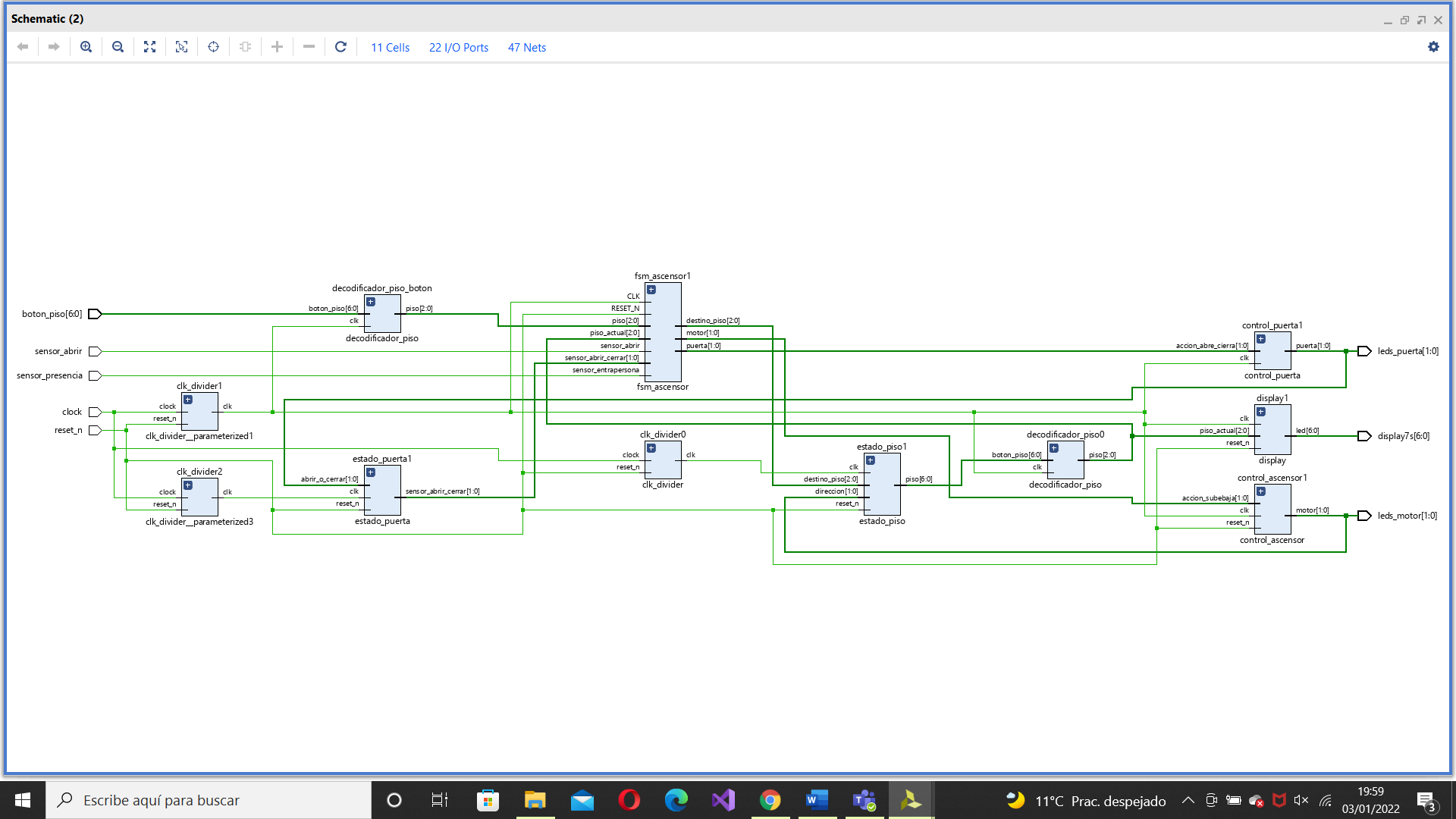
Esto lo haremos a través de simulaciones de cada componente que añadiremos a la rama test en el control de versiones de github.

* Cierre de pre-síntesis

Una vez hayamos simulado todos los componentes y hayamos visto que no hay errores ni sintácticos ni de síntesis podremos seguir desarrollado el diseño.

* Síntesis

Este proceso es fundamental ya que sirve para refinar y depurar el proceso para que quede lo más limpio y correcto posible.



* Proceso de verificación post-síntesis

En nuestro caso es inmediato ya que como no hemos añadido nuevas restricciones después de la síntesis la verificación es efectiva de inmediato.

* Cierre de diseño

Por último, podemos declarar el cierre del diseño ya que:

1. El diseño cumple todas las restricciones (en nuestro caso se limite a las únicas que tenemos que son las iniciales).
2. Se cumplen las especificaciones temporales.

**Problemas ocasionados**

1. El primer problema con el que nos encontramos fue como podríamos hacer que cada distintas partes del diseño funcionasen a frecuencias diferentes si cogíamos la misma señal del reloj de la placa.

Tras búsqueda en internet y en los apuntes vimos que podríamos hacer una división de la frecuencia del reloj original de la placa adecuada a la necesaria para cada dispositivo.

Como tendríamos que realizar este proceso varias veces vimos que la mejor manera era realizando un componente cuya única función fuese dividir la frecuencia del reloj de la placa, para que sirviese con distintas frecuencias a distintos módulos o dispositivos.

**2.**  Sincronización de control puerta y estado puerta al igual que control ascensor y estado piso. Aunque inicialmente parecía que sus funcionalidades estaban muy definidas, al ir implementando todos los módulos, se fue disipando esa claridad de ideas y pusimos demasiado peso en ambos módulos y se pisaban la funcionalidad entre ellos.

Esto lo pudimos solucionar dejando que los módulos de control se limitasen solamente a pasar información de la fsm a los módulos de estado que eran los que de verdad manejaban las salidas de motor y puerta.

**Conclusión**

Al principio intentamos abordar este trabajo con un método o forma botton-top en el cual intentábamos hacer cada módulo desde cero para luego realizar la parte estructural juntando todos los módulos y tener la top completa. Gracias a las tutorías vimos que este método era erróneo ya que perdíamos la claridad de la idea intentando realizar todo a la vez desde un principio, además de que esto ocasionaba muchos errores.

Finalmente optamos por el método top-botton en el cual primero meditamos bien que íbamos a realizar y que requisitos necesitábamos, luego casi todo el trabajo fue la separación de cada tarea en sus bloques y ver como podíamos minimizar al máximo la influencia de la fsm( ya que en un principio todo lo hacia este módulo). De esta forma iniciamos haciendo el diagrama de bloques y viendo la funcionalidad de cada uno, posteriormente hicimos unos testbench muy sencillos para cada funcionalidad pensada para ellos y así finalmente realizamos el código de cada módulo de manera mucho más sencilla.

En conclusión, merece mucho la pena en destinar mucho tiempo solo a pensar y tener claros los requisitos y las funcionalidades porque luego esto facilitará infinitamente el trabajo de codigo.